

Optimization of reliability and speed of the end-of-line quality inspection of electric motors using machine learning

Mlinarič, J.^{a,b,*}, Pregelj, B.^a, Boškosi, P.^a, Dolanc, G.^a, Petrovčič, J.^a

^aJozef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia

^bJozef Stefan International Postgraduate School, Ljubljana, Slovenia

ABSTRACT

Consistently maintaining high-end product quality in the production process is challenging. End-quality inspection must be highly sensitive to detect even minimal deviations, while being fast and accurate. However, quality inspection systems often face calibration intricacies, are time-consuming, and rely heavily on expert knowledge. They handle substantial data flows and inspect numerous features, some of which contribute minimally to the final grade. To address these challenges, the paper proposes employing statistically supervised machine learning methods for classification. Decision trees, Random forests, Bagging, and Gradient boosting classifiers are recommended for feature selection and accurate diagnosis, particularly for electric motor classification. By utilizing the feature importance attribute for feature selection, the proposed approach compares model accuracies, reducing ramp-up and commission times significantly. The study found that all suggested classifiers achieved high accuracy in classifying electric motors in end-of-line quality inspection system. Moreover, they effectively reduced the number of features and optimize database operations. Utilizing a reduced feature set streamlined diagnostic algorithms, accelerated learning, and improved model interpretability, enhancing overall efficiency and comprehension. Furthermore, analysing the feature importance attribute could simplify diagnostic hardware and expedite quality inspection by eliminating unnecessary steps. Newly generated models can also verify expert decisions on feature selection and limit adjustments, enhancing efficiency in production processes.

ARTICLE INFO

Keywords:
Quality inspection;
Fault detection;
Machine learning;
Feature selection and classification;
Feature importance;
Decision trees;
Random forests;
Bagging;
Gradient boosting algorithm

**Corresponding author:*
jerne.mlinaric@ijs.si
(Mlinarič, J.)

Article history:
Received 20 February 2024
Revised 8 May 2024
Accepted 27 May 2024



Content from this work may be used under the terms of the Creative Commons Attribution 4.0 International Licence (CC BY 4.0). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

1. Introduction

Electric motors are one of the most mass-produced devices and they are produced at highly automated manufacturing lines equipped with 100 % end-of-line (EoL) quality inspection, [1, 2]. Maintaining constant product quality, detecting faulty products and preventing them to be delivered to the customers or further built into devices and systems is highly important. With each production/integration step of faulty part, replacement costs increase substantially [1]. For this reason, a great attention is put to the design and implementation of fully automated EoL quality inspection systems. These must be reliable and at the same time fast enough not to hinder the production line pace. In the presented case, quality inspection of electric motors is performed in a non-invasive way, where several variables are measured during short test run of the motor. The measured variables comprise electric parameters (voltage, current and power), speed, torque, vibrations at different points of motor body and sound at different rotational speeds [1]. The

mentioned signals are sampled by high frequency (e.g. 50 kHz) and further processed by signal processing methods (e.g. Digital Filtering, Fast Fourier Transformation, etc.). This reduces the amount of data considerably while preserving relevant information, but still results in a high number of calculated parameters – features, representing basis for motor fault detection and isolation. High number of features can be impractical from several reasons: a) not all features carry useful information, b) some feature may carry the same information, c) comparing a high number of features against their thresholds values may be time consuming and finally, d) determining (learning) feature threshold values is demanding and time-consuming process.

With an ultimate goal of developing reliable and fast quality inspection methods, this paper deals with the problem of reduction of feature space to a limited subspace of relevant features, carrying enough information for motor quality inspection. The space of features can be reduced by implementing machine learning methods which select only the relevant features. Since not all relevant features contribute the same amount of information to final classification, additionally evaluation of feature selection describes the influence of each observed feature. Therefore, the features with the minimal influence can be eliminated from learning procedure. This can significantly decrease computational demand during learning (ramp-up and commission time) and operation phases. Moreover, in certain cases it can even lead to elimination of particular measurements (sensors), thus simplifying the inspection system hardware and software as well as speeding-up the EoL testing procedure. Study presented in this paper is based on the real industrial data derived from real EoL quality inspection systems installed at the production site of one of renowned European mass producer of electric motors. EoL quality inspection line, which is subject of this paper, was designed and implemented by the authors of the paper.

The paper is organized as follows: In Section 2, the subject of inspection and existing quality inspection procedure/system are briefly described. The structure of the measured data record and resulting feature set generated by the inspection of one motor is described. This is then followed by the Section 3, where machine learning algorithms (Decision tree, Random forest, Bagging and Gradient boosting) for feature selection are presented. In Section 4 the presented algorithms are evaluated and compared.

2. Problem description: Subject of inspection and quality inspection system

The subjects of inspection are brushless DC (BLDC) motors for domestic and automotive applications. An example of such motor used for vacuum cleaning applications, is shown on the Fig. 1. The addressed motors are manufactured by the renowned mass producer (Domel Slovenia, [3]). The production takes place at fully automated assembly line equipped with modular EoL quality inspection system, presented on the Fig. 2. More details can be found in [2], which describes similar system.

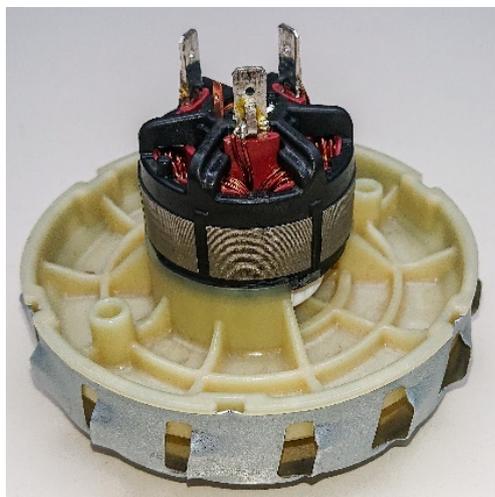


Fig. 1 Example of the BLDC motor (subject of EoL quality inspection)



Fig. 2 Modular EoL quality inspection system

The EoL quality inspection system assures “100 % quality inspection” meaning that each produced motor undergoes the test procedure. In general, the faults fall in two categories: electrical and mechanical. The latter can be further divided to rotor, bearing and turbine faults. Rotor and bearing faults are comprehensively elaborated in [4] whereas the explanation of turbine faults can be found in [1, 2]. The main steps of quality inspection procedure are described in following sections 2.1, 2.2 and 2.3 and in the Fig. 3.

2.1 Measurement and data acquisition

In the EoL quality inspection system, each motor is started several times and during short test runs various motor parameter are measured by the automatic measuring and data acquisition system (Fig. 3, square 1). The following parameters are measured by sensors: electric parameters (winding voltage and current, power, power supply current and voltage), vacuum pressure, rotation speed, vibrations at several points of motor body, sound at low and high rotational speed and also environmental conditions (ambient temperature and pressure) to compensate their effect on motor performance. The results of measurements are time series (waveforms) of particular parameter, and they represent “raw signals”. Depending on observed parameter and derived features, signals are acquired at specific sampling frequency (typically 10-60 kHz) and measurement duration (typically from 0.1 up to 1 s), resulting in timeseries of various lengths (from 1000 samples to 30000 samples).

2.2 Feature extraction by signal processing

To reduce the amount of data and to extract the relevant information, raw signals are processed by signal processing methods, such as filtering (low-pass, high-pass, band-pass filters), down-sampling, averaging, frequency analysis, etc. The outcome of signal processing is a set of “features”, which are detailed in [4] and shown on the Fig. 3 (square 2). They are in general:

- Root-Mean-Square (RMS) values of band-pass-filtered waveforms;
- Power of signals at particular frequencies;
- Aggregated/actual values obtained from specific measurement equipment.

Details of feature extraction and signal processing algorithm are not described in this paper as they are subject of past research and development, elaborated in detail in [1, 5]. In this particular case the signal processing algorithm generates 80 features, where each feature is represented by floating-point numeric value.

2.3 Diagnostic result generation

Based on the on the values of the features, final diagnostic result of the inspected motor is generated by simple rules, as follows from the Table 1. For each feature it is checked if it is inside specified range.

Table 1 Diagnostic result generation

Measurements	Features	Diagnostic result
Completed	All features are within specified ranges	Motor GOOD
	One or more features are outside specified range	Motor BAD
Not completed, due to:		UNDEFINED
• measurement faults		
• sensor faults		
• motor manipulation fault		
• motor transport faults		
• etc.		

At the start of the production of new motor type, the motors from test-production set (series 0) are assessed as good or bad by the skilled experts. Based on experiences, the experts select features that are going to be used in diagnostic result generation. For this case-study system and motor type, 36 of total 80 features were chosen, and for each of chosen features two limit values (low and high) are set. In practice, all this is done manually by skilled experts. This is time-consuming and highly depends on expert skills. In addition, this method requires regular updates and fine-tuning of limit values when the mass production starts and production volumes increases [4].

Fig. 3 illustrates the entire procedure of measurement and data acquisition, feature extraction and diagnostic result generation. The whole process for one motor can be executed under 30 s, but due to parallel execution of diagnosis steps, the motor inspection rate is 10 s, which means that every 10 s one motor exists the EoL inspection system.

The described quality inspection algorithm successfully detects motors with insufficient quality, but it has some drawbacks:

- High number of original features (80) leads to high number of feature range limits that must be defined;
- Some features carry similar information (redundancy);
- Some features carry no useful information;
- Skilled expert is required to remove redundant features and features that carry no relevant information and to adjust limit values of feature in use. This is difficult and time consuming and becomes an issue during start of production and commissioning of new motor types.

Based on that there are four main goals of the study presented in this paper:

- Automatic selection relevant of features (removing redundant features and features that do not carry relevant information);
- Decrease the dependence on human expert skills;
- Automatic determination of feature limit values;
- Generating classification models with set of features that hold 95 % of useful information;
- Reducing ramp-up and commission time of the quality inspection system.

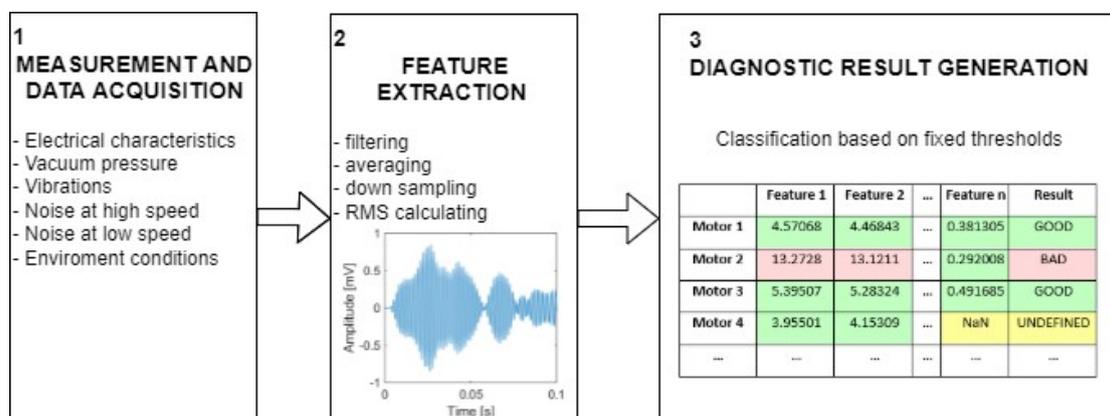


Fig. 3 Data transformation during the procedure

3. Feature selection using machine learning methods

In this section, dedicated machine learning methods will be used to automatically select the relevant features and set their threshold values. In general, feature selection is an effective way to deal with dimensionality [6] and it is often used in areas where a huge amount data is being obtained, such as identifying genes [7-9], image classification and analysing [10-13] text classification [14-16], in recent times also for monitoring manufacturing processes and quality control [17-19]. Feature selection aims to identify and retain the most relevant features while discarding redundant or uninformative ones by determining the “degree of usefulness” of a specific feature. By reducing the number of features, also the number feature limit values to be adjusted is reduced. The risk of the associated quality inspection errors [6] is also decreased. However, within the set of informative features, some may be significantly more informative than others.

The methods presented not only eliminate non-informative features but also sort the remaining features according to their informativeness. The goal of this paper is therefore to assess whether the quality inspection can be successfully performed by using only the limited number of the most informative features. Feature selection reduces the dimensionality of the data; therefore, data mining algorithms can be operated faster and more efficient [6]. Reduced amount of input data simplifies the interpretability of tree-like machine learning methods [20]. Additionally, such simplified classification methods and reduced input datasets decrease the ramp-up and commission time of quality inspection system and whole production line. Since some redundant and non-informative features are removed, sensors associated with removed features can potentially be eliminated. Optimization and reordering of the diagnostic steps based on feature importance can speed-up the quality inspection procedure.

3.1 Supervised machine learning classification methods

Supervised machine learning methods were selected since the labelled data for learning is available. These methods offer several advantages, including high reliability based on statistics, robustness, and reduction of the need for expert knowledge (e.g. knowledge about physical background of the system). However, in order to establish a supervised learning method, a sufficient amount of data from the production process is required. Therefore, these methods are suitable for manufacturing lines for mass production, like the one presented in this paper, where a lot of data is generated. In this paper 4 different methods were tested and compared:

- Decision tree classifier (DT);
- Random forest classifier (RF);
- Bagging classifier (BG);
- Gradient boosting classifier (GB).

The decision tree classifier partitions the instance space through a recursive process, forming a tree model where top nodes (roots) lack incoming edges, while internal nodes (test nodes) split the space based on attribute values. Internal nodes symbolize decision points, and bottom nodes (leaves) indicate decision outcomes [21, 22].

Random forests is a powerful ensemble learning method that combines multiple tree predictors. It belongs to family of averaging methods, meaning, the driving principle is to build several estimators independently and then to average their predictions [23-25].

Bagging, short for bootstrap aggregating, is a powerful and straightforward method for constructing an ensemble of classifiers. It also belongs to family of averaging methods. It combines multiple classifiers' outputs for improved accuracy by training each on a subset of instances randomly drawn from the training set [22, 26, 27].

Gradient boosting is a technique for improving the performance of weak learners [28]. It belongs to the family of boosting methods, meaning, base estimators are built sequentially, and one tries to reduce the bias of the combined estimator. It enhances weak learners sequentially, aiming to reduce the combined estimator's bias. The technique combines weak models for a powerful ensemble, particularly effective in decision trees [22, 25].

While the decision tree classifier (DT) stands alone, the remaining three classifiers (RF, BA,

GB) belong to the ensemble-based methods category. All methods here are “tree-like” classifiers and can be represented as decision trees using IF-THEN rules. The methods automatically generate IF-THEN rules for classification, including threshold values for each observed feature.

3.2 Data for learning and evaluation

To implement and test the methods, the data is needed. Data was generated by existing EoL quality inspection system (Fig. 2) during inspection of a total of 37440 motors. Generated data contains raw time series of measured signals and extracted features (mentioned 80 features), as follows from the Section 2. For machine learning algorithms, all features are used. The whole data set of features can be presented as 37440×80 matrix. The quality inspection results (1=Motor GOOD, 2 = Motor BAD, 0 = UNDEFINED) represents 37440×1 vector. The entire data set was divided into two parts: training data (75 % of all data) and testing data (25 % of all data). The situation is represented by the Table 2.

The data records of all 37440 motors were randomly distributed between training data and testing data to compensate for possible time drift of product quality. The same training and testing datasets were used during the training and testing of all 4 machine learning methods. Dataset remained unchanged throughout the entire training process. All involved features are named by symbols and anonymized to prevent the disclosure of sensitive technical data.

Table 2 Training and testing data arrangement

	F1	F2	...	F80	R	
M1	X	X	X	X	X	Training data (75 %)
M2	X	X	X	X	X	
...	X	X	X	X	X	
M28080	X	X	X	X	X	
M28081	X	X	X	X	X	Testing data (25 %)
...	X	X	X	X	X	
M37440	X	X	X	X	X	

M – motors, F – features, R – quality inspection results

3.3 Implementation

Fig. 4 illustrates the proposed process of feature selection.

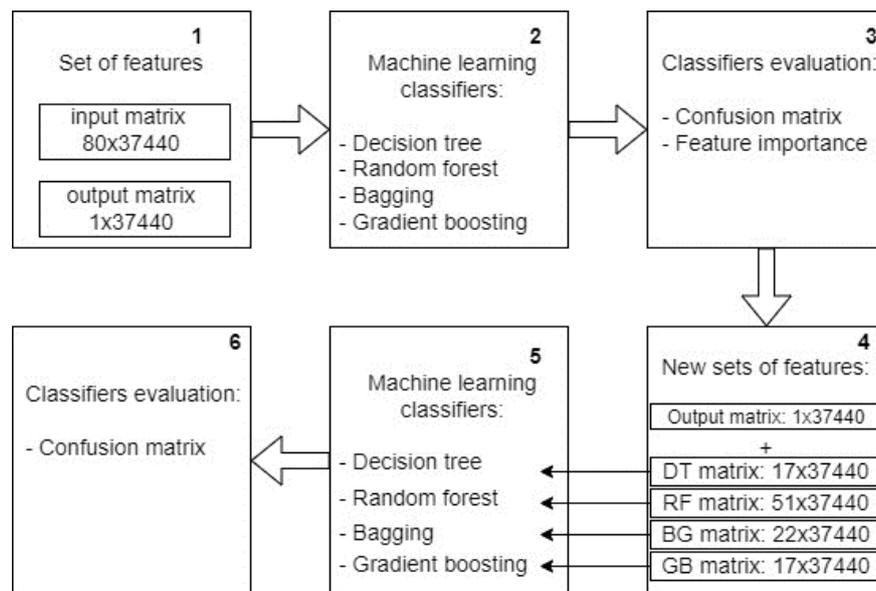


Fig. 4 Flow chart of the procedure

The process is carried-out in six steps:

1. Generation of original feature set (an input matrix of size 80 features × 37440 motors and output matrix of size one feature – grade × 37440 motors) obtained by data acquisition and signal processing, described in Section 2.
2. First machine learning to train the classifiers. Feature selection is utilized to eliminate redundant features. Depending on the type of classifier, the range of selected features is reduced to 38 (DT classifier), 78 (RF classifier), 42 (BG classifier), and 35 (GB classifier).
3. Evaluation of trained classifiers to present their performance and capability to evaluate features' classification impact and importance. From obtained results it followed:
 - a. Set of features with 95 % informativeness additionally reduce space of features. Depending on the type of classifier, the range of selected features is additionally reduced to 17 (DT classifier), 51 (RF classifier), 22 (BG classifier), and 17 (GB classifier).
 - b. Certain features, despite their low importance, still persist. Therefore, it was decided to check the performance of reduced classifiers with features that contains 95 % of all useful information.
4. Generation of reduced sets of training and testing data with features with 95 % of useful information for each classifier, output matrix remains the same from step 1.
5. Second machine learning to train the classifiers with new reduced dataset.
6. Evaluation of new classifiers and comparison to the results of classifiers from step 2.

All classification methods were implemented and tested in Python using the *scikit-learn* (sklearn) library. Cross-validation was employed during training (step 2 and step 5) to ensure robust model evaluation and to prevent model overfitting. For visualization and data manipulation, *matplotlib*, *numpy*, and *pandas* libraries were also utilized.

3.4 Presentation of the results and comparison of the methods

Following the training phases, the trained algorithms were evaluated using the testing data, and the predicted output classes (quality inspection result) were compared to the actual output classes. For each method, outcomes are presented in the form of well-known Confusion Matrix (CM). The CM provides numerical and visual representation of the classification algorithm's accuracy. It consists of columns representing the predicted output classes and rows representing the actual output classes. In the presented case, since there are three classes, the size of the CM is 3 × 3 as shown in the Table 3. The diagonal elements represent correctly classified instances, while the off-diagonal elements represent miss-classified instances.

Table 3 Confusion matrix structure

		PREDICTED		
		UNDEFINED	Motor BAD	Motor GOOD
ACTUAL	UNDEFINED			
	Motor BAD			
	Motor GOOD			

The CM provides valuable information about the miss-classification, but it does not directly capture the cost associated with each type of miss-classification. To address this, the Miss-classification cost matrix can be introduced. This matrix assigns specific costs to different types of miss-classifications, considering the relative importance or impact of miss-classifying different classes. The Miss-classification cost matrix has the same dimension as the Confusion matrix and consists

of numerical values, as shown in the Table 4. The diagonal elements of the Miss-classification cost matrix are set to zero (correct classification represent no cost). The off-diagonal elements represent the costs associated with miss-classified instances, with larger values indicating higher risks or costs associated with miss-classification. In our case the highest cost is associated with situation when actual faulty or undefined motor is recognized as good one and delivered to the customer (actual class BAD or UNDEFINED, predicted class GOOD). The cost of this miss-classification is set to 10. Bad motor that is predicted as undefined (and the opposite) presents low cost of miss-classification, therefore these cases are graded with 0.5. Good motor, ranked as bad or undefined, does not present any risk for the customer, but it represents an unnecessary waste of motors, therefore it is marked with 1.

To calculate the overall miss-classification cost, the CM the Miss-classification cost matrix are multiplied (element-by-element multiplication) resulting in new 3×3 matrix. Total cost is calculated by adding up all 9 elements of resulting matrix. This value provides a measure of the total cost incurred due to miss-classification. Ideally, a well-performing classifier would have a miss-classification cost close to zero, indicating minimal miss-classification and associated risks.

The accuracy of a classifier is a measure of its performance and is calculated as the ratio between the number of correctly classified elements and the total number of elements. The desired accuracy is close to 1 (100 %), indicating that almost all elements were classified correctly.

Table 4 Cost matrix

		PREDICTED		
		UNDEFINED	Motor BAD	Motor GOOD
ACTUAL	UNDEFINED	0	0.5	10
	Motor BAD	0.5	0	10
	Motor GOOD	1	1	0

4. Analysis of the results

During machine learning, all methods generated own IF-THEN rules for classification. In 3.1 it is explained that chosen methods are “tree-like” and can be explained with IF-THEN rules. In the Fig. 5, an example of the decision tree of the decision tree classifier is presented. The figure shows a diagnosis procedure with a tree-like set of rules and leaves. Since the generated decision tree is very extensive, one rule for the first branch is explained. At the enlarged part of the figure, the auto generated threshold value for one particular feature (BE_H2) is illustrated. At enlarged part the gini index value [6, 21, 29] is also illustrated and used as splitting criteria [6]. Further, at this branch 27064 samples are involved in classification process, where 600 of them are marked as UNDEFINED, 1355 as Motor BAD and 25109 as Motor GOOD.

Each of the methods generates similar decision tree scheme where rules are defined by observed features and their threshold values. Threshold values are set automatically during machine learning and since they are presented as real values. They can be easily checked and re-adjusted. Such examination of the decision tree structure of each classifier enhances comprehension of the decision-making process employed by each method. This understanding is particularly valuable in industrial applications where interpretability is paramount, as it allows users to gain insights and interpret the decision-making process with clarity.

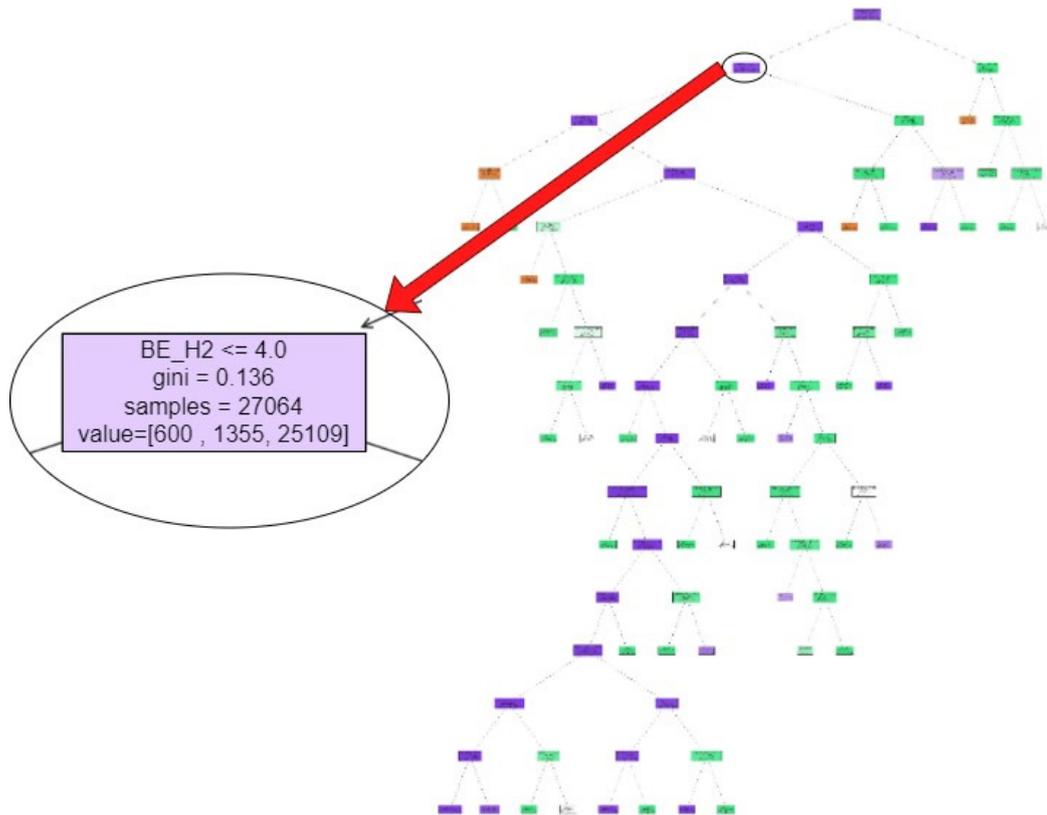


Fig. 5 Decision tree of the decision tree classifier

Results of feature selection are collected in Table 5, where the comparison between classifiers is presented. The results show that all methods successfully executed feature selection in the step 2 (Fig. 4). The *gradient boosting* classifier selected the lowest number of important features, meaning, this method requires the smallest amount of data for successful classification. The least successful method here is *random forest*.

However, the best performance shows the classifier with the lowest number of wrong classified motors and with the lowest miss-classification cost. Therefore, *bagging* overperformed all other classifiers and *gradient boosting* yield the worse performance. Fig. 6, Fig. 8, Fig. 10 and Fig. 12 present confusion matrices for each classifier, generated at step 2.

Table 5 Comparison of observed classifiers

	Decision tree	Random forest	Bagging	Gradient boosting
No. of selected features with original dataset after step 2	38	78	42	35
Number of features for 95 % informativeness	17	51	22	17
Influence of 10 most important features	92 %	69 %	88.6 %	91.4 %
Accuracy of classifiers with original dataset (step 3)	99.2 %	99.33 %	99.46 %	99.19 %
Accuracy of classifiers with reduced dataset (step 6)	99.16 %	99.33 %	99.46 %	99.16 %
No. of wrong classified motors of classifiers with original dataset (step 3)	75	53	51	76
No. of wrong classified motors of classifiers with reduced dataset (step 6)	79	53	51	79
Miss-classification cost of classifiers with original dataset (step 3)	426	477	312	679
Miss-classification cost of classifiers with reduced dataset (step 6)	430	468	348	637

Further insights into feature selection across different classifiers show that the most influential features hold the majority of information, useful for classification. Third row in the Table 5 shows a part of information contained in the 10 most important features of each method. In all methods (except RF) top ten features contain the majority of information and to achieve 95 % of data informativeness, 17-51 features are required. This comprehensive evaluation provides insights into the strength and weakness of each classifier, considering both accuracy and miss-classification cost. At the Table 6, where the 10 most influential features for each classifier in the order of importance are listed, it is shown that all classifiers recognize the majority of features as important (Figs. 14-17 visually illustrate the informativeness of each observed classifier for 10 most important features). At the Table 7 all features involved in machine learning are listed and in columns each of them is marked whenever it appears to be important for each classifier. With gray are coloured rows of features that are important for all observed classifiers.

These findings can be highly beneficial for tuning classification parameters. Instead of adjusting the thresholds for all observed features, only the most influential features need to be addressed. Additionally, since some features do not contribute to the final classification decision, they do not need to be stored in the company database. This results in the reduced computational burden by focusing only on the most informative features, a smaller data flow between the local computer and the company database, reduces the potential for communication errors, and ultimately requires less storage space.

Table 6 List of 10 most important features for each classifier

	Decision tree	Random forest	Bagging	Gradient boost
1. important feature	BE_H1	BE_H2	BE_H1	BE_H1
2. important feature	BE_H2	BE_H1	BE_H2	BE_H2
3. important feature	BE_H3	BE_H3	BE_H5	VRC
4. important feature	VA	BE_H5	VRC	VA
5. important feature	VRC	BE_H4	VA	VA_H1
6. important feature	HW_H1S	BE_H6	HW_H1S	HW_H1S
7. important feature	VRL_H2	VRC	VA_H1	BE_H3
8. important feature	HW_H6	VA	BE_H3	BE_H6
9. important feature	AVR_V	HW_H1S	BE_H6	VRL_H2
10. important feature	VW_V	VA_H1	HW_H6	FR_H2

Table 7 All features involved in machine learning and their recognition as important for each classifier

Feature name	DT	RF	BG	GB
VA	X	X	X	X
VA_H1		X	X	X
VA_H2		X		
VA_H3		X		
VA_H4		X		
VA_H5		X		
VA_H6	X	X		
VA_H7		X		
VA_H8		X		
VA_H9		X		X
VA_H10		X		
VA_H11		X		
VA_H12		X		X
VA_H13	X	X		
VA_H14		X		
VA_H15	X	X		
AVR_U		X	X	X
AVR_V	X	X	X	X
AVR_W		X		
BM_U				X
BM_V		X	X	X
BM_W		X	X	
FR_H1		X	X	X
FR_H2		X	X	X
BE_H0				
BE_H1		X	X	X
BE_H2		X	X	X
BE_H3		X	X	X
BE_H4		X	X	X
BE_H5		X	X	X
BE_H6		X	X	X
CP		X	X	X
VW_U			X	X
VW_V		X	X	X
VW_W		X	X	X
VRC		X	X	X
VRC_H1			X	
VRC_H2		X	X	
VRC_H3			X	
VRC_H4		X	X	X

VRC_H5		X		X
VRC_H6		X		
VRC_H7		X		
VRC_H8		X		
VRC_H9	X	X	X	
VRC_H10		X		
VRC_H11		X		X
VRC_H12	X	X		
VRC_H13		X	X	X
VRC_H14		X		X
VRC_H15		X	X	X
VRL		X		
VRL_H1		X	X	
VRL_H2	X	X		X
VRL_H3		X		
VRL_H4		X		
VRL_H5		X	X	X
VRL_H6	X	X	X	X
VRL_H7		X	X	
VRL_H8		X	X	

VRL_H9	X	X	X	X
VRL_H10		X	X	X
VRL_H11		X		
VRL_H12		X	X	
VRL_H13	X	X		
VRL_H14		X		
VRL_H15		X		
MV	X	X	X	X
CW_U	X	X		
CW_V	X	X	X	
CW_W	X	X	X	
HW_H1S	X	X	X	X
HW_H18	X	X	X	X
HW_H2	X	X	X	X
HW_H2S	X	X	X	X
HW_H3	X	X	X	X
HW_H4	X	X	X	X
HW_H6	X	X	X	X
HW_H9		X	X	
REV				

At last, the performance of classifiers, trained with features of 95 % information (step 5), is evaluated (step 6). The result in Table 5 shows that the accuracies and performances of classifiers do not change a lot. However, despite the RF and GB classifier, the cost of miss-classification increased, meaning, those classifiers miss-classified performed slightly worse. (as shown at Fig 6-13). Despite minor fluctuations in accuracy, the overall performance remains relatively stable what indicate the robustness of chosen methods. However, the increase in miss-classification costs for certain classifiers indicates potential areas for optimization in future iterations.

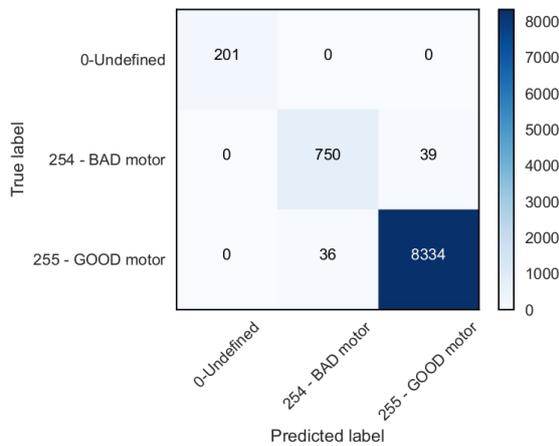


Fig. 6 Confusion matrix for decision tree classifier (38 features)

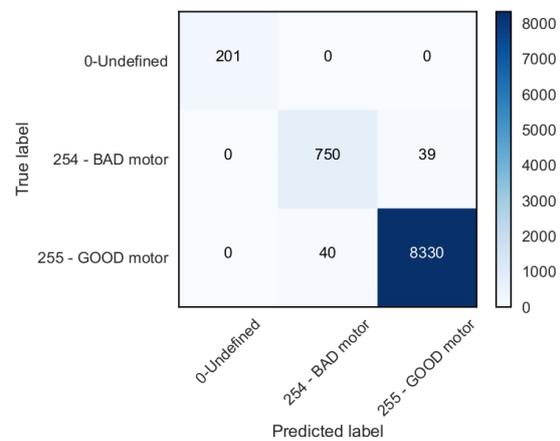


Fig. 7 Confusion matrix for decision tree classifier (17 features)

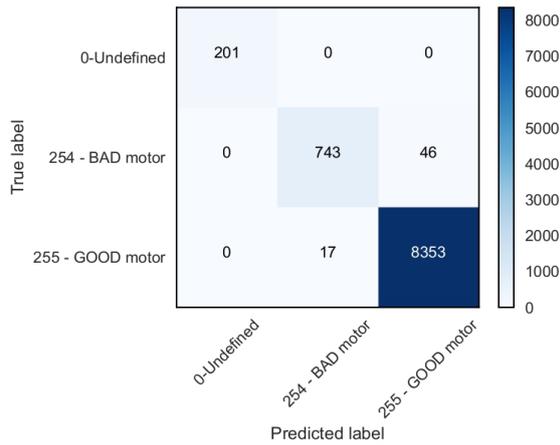


Fig. 8 Confusion matrix for random forest classifier (78 features)

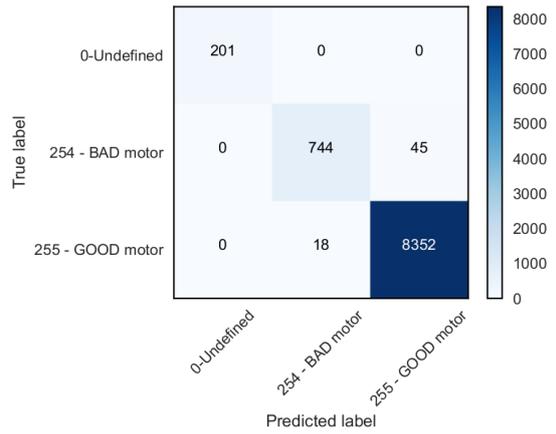


Fig. 9 Confusion matrix random forest classifier (51 features)

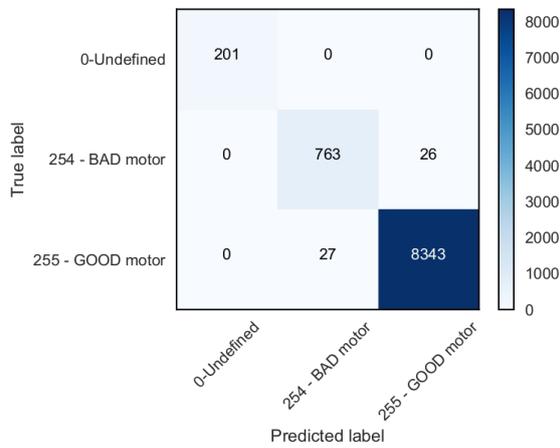


Fig. 10 Confusion matrix for bagging classifier (42 features)

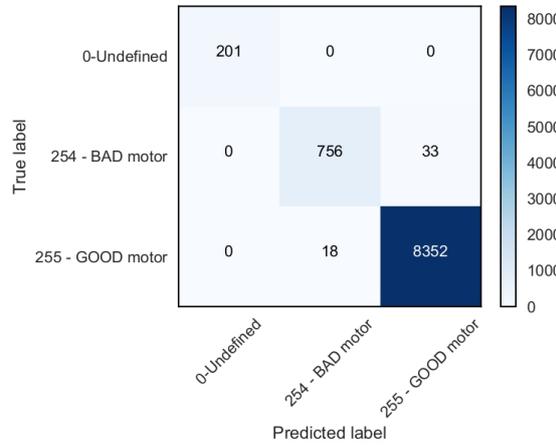


Fig. 11 Confusion matrix for bagging classifier (22 features)

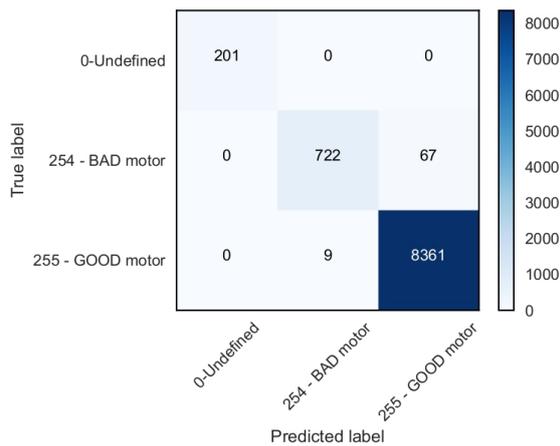


Fig. 12 Confusion matrix for gradient boost classifier (35 features)

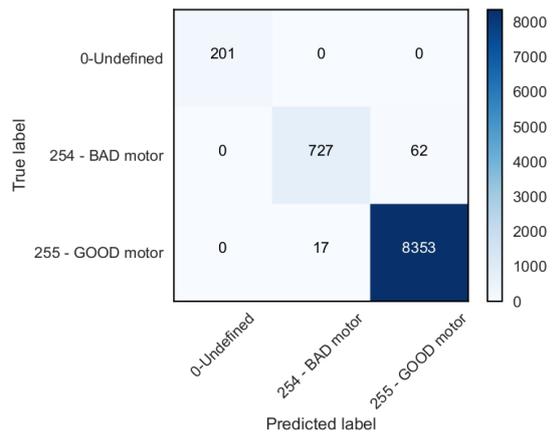


Fig. 13 Confusion matrix for gradient boost classifier (17 features)

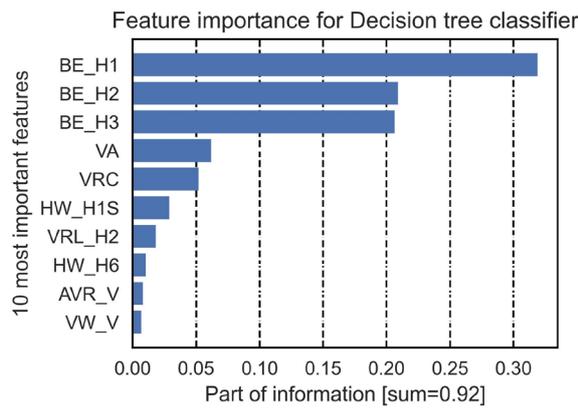


Fig. 14 Feature importance for decision tree classifier

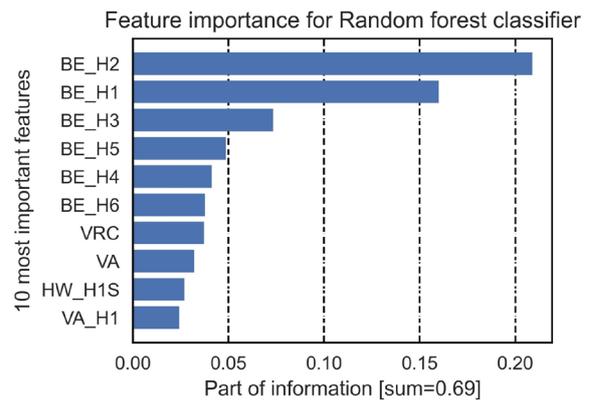


Fig. 15 Feature importance for random forest classifier

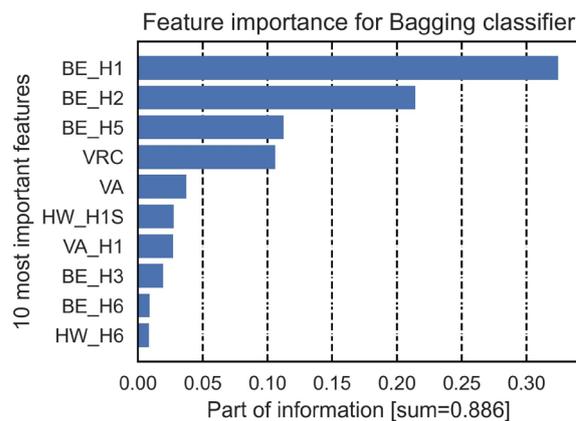


Fig. 16 Feature importance for bagging classifier

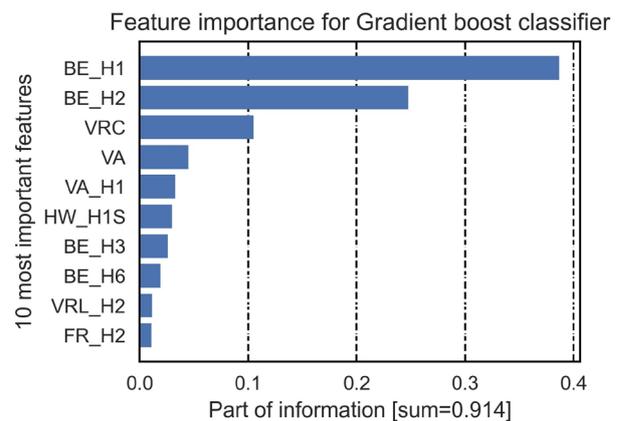


Fig. 17 Feature importance for gradient boost classifier

5. Conclusion

This study introduced and compared various classifiers for feature selection purposes used for automated end-of-line quality inspection of electric motors within the real manufacturing line. Decision tree, Random forest, Bagging, and Gradient boosting classifiers were implemented and assessed based on their complexity (number of selected features), accuracy, and the impact of the important features. Initial goal of the study was achieved successfully. All four tested classifiers demonstrated high accuracy, proving their suitability for electric motor classification in an end-of-line quality inspection system. All investigated classifiers successfully reduced the number of features and thus optimized the database operation. Further, the second step of feature selection, with a reduced dataset featuring features that hold 95 % of useful information, yielded high accuracy of trained classifiers. This reduced feature set simplifies the diagnostic algorithm, speeds-up its' learning, improves the interpretability of the observed models and makes them more understandable and explainable. In addition, new classification models, learned with reduced dataset, simplify the end-of-line quality inspection, decrease the ramp-up and commission time, eliminate unnecessary steps of diagnosis, reduce equipment complexity (in some cases eliminate the need for particular sensors), reduce costs, and minimize data flow. Consequently, company databases are optimized. Due to fully automated learning procedures, reliance on specialized experts is reduced. Developed classification models can also be used as a verification of experts' decision regarding feature selection and threshold values adjustment. In summary, this study encompasses insights into feature selection, practical implications for industrial applications considering methods robustness and comprehensive evaluation of different classifier, considering accuracy and miss-classification cost, aiding in decision making when selecting the most suitable classifier for specific application.

During research and implementation, two interesting and useful future topics of research were identified: 1. Transferability of classification models, and 2. Condition monitoring of production lines.

Within the first topic it could be investigated if classification models derived for one motor type could be used for quality inspection of similar motor types, or if they could increase learning procedure of new motor types. Based on the manufacturer's experiences, the diagnosis procedure across various manufacturing lines follows a similar approach, leading to the detection of similar faults and malfunctions across different products. Furthermore, common features are identified, suggesting that the feature selection process from observed classifiers of a particular motor type could be applied to other product. This transferability would be beneficial especially for small-series products. Some motor types are manufactured in small-series (e.g. up to thousand pieces per year), therefore it is challenging to establish accurate classification model with limited amount of learning data. Exploring the applicability of these findings across various motor types can speed-up the creation of quality inspection algorithms for new motor types in the future. As new motor types are developed frequently and produced in varying quantities, the transferability of the methods can establish a standardized approach to implement quality inspection algorithms for new motor types, reducing costs and thus enhancing the whole manufacturing process.

Second topics regards possibilities of condition monitoring of production line. In normal conditions (when there is no degradation of the manufacturing process) features importance attributes do not significantly change with time. On the other hand, an increase of importance of particular feature may indicate the issue of particular manufacturing operation or an issue of input material or components. Periodic evaluation of feature importance attributes can therefore help to detect faults or degradations of various steps of manufacturing process or issues with input materials and components.

Acknowledgments

This work was supported by the Slovenian Research Agency under Grant P2-0001; Slovenian Research Agency under Grant L2-4454.

References

- [1] Juričić, Đ., Petrovčič, J., Benko, U., Musizza, B., Dolanc, G., Boškoski, P., Petelin, D. (2013). End-quality control in the manufacturing of electrical motors, In: Strmčnik, S., Juričić, Đ. (eds.), *Case studies in control, Advances in industrial control*, Springer, London, United Kingdom, 221-256, doi: [10.1007/978-1-4471-5176-0_8](https://doi.org/10.1007/978-1-4471-5176-0_8).
- [2] Benko, U., Petrovčič, J., Musizza, B., Juričić, Đ. (2008). A system for automated final quality assessment in the manufacturing of vacuum cleaner motors, *IFAC Proceedings Volumes*, Vol. 41, No. 2, 7399-7404, doi: [10.3182/20080706-5-KR-1001.01251](https://doi.org/10.3182/20080706-5-KR-1001.01251).
- [3] Domel, Domel d.o.o., from <https://www.domel.com/sl>, accessed September 26, 2023.
- [4] Boškoski, P., Petrovčič, J., Musizza, B., Juričić, Đ. (2011). An end-quality assessment system for electronically commutated motors based on evidential reasoning, *Expert Systems with Applications*, Vol. 38, No. 11, 13816-13826, doi: [10.1016/j.eswa.2011.04.185](https://doi.org/10.1016/j.eswa.2011.04.185).
- [5] Benko, U., Petrovčič, J., Juričić, Đ. (2005). In-depth fault diagnosis of small universal motors based on acoustic analysis, *IFAC Proceedings Volumes*, Vol. 38, No. 1, 323-328, doi: [10.3182/20050703-6-cz-1902.01856](https://doi.org/10.3182/20050703-6-cz-1902.01856).
- [6] Rokach, L., Maimon, O. (2014). *Data mining with decision trees, Theory and applications*, 2nd Edition, World Scientific, New Jersey, USA, doi: [10.1142/9097](https://doi.org/10.1142/9097).
- [7] Kim, S., Xing, E.P. (2009). Statistical estimation of correlated genome associations to a quantitative trait network, *PLOS Genetics*, Vol. 5, No. 8, Article No. e1000587, doi: [10.1371/journal.pgen.1000587](https://doi.org/10.1371/journal.pgen.1000587).
- [8] Beisvag, V., Jünge, F.K.F., Bergum, H., Jølsum, L., Lydersen, S., Günther, C.-C., Ramampiaro, H., Langaas, M., Sandvik, A.K., Lægred, A. (2006). GeneTools - application for functional annotation and statistical hypothesis testing, *BMC Bioinformatics*, Vol. 7, No. 1, Article No. 470, doi: [10.1186/1471-2105-7-470](https://doi.org/10.1186/1471-2105-7-470).
- [9] Kuehl, P.M., Weisemann, J.M., Touchman, J.W., Green, E.D., Boguski, M.S. (1999). An effective approach for analyzing "prefinished" genomic sequence data, *Genome Research*, Vol. 9, No. 2, 189-194, doi: [10.1101/gr.9.2.189](https://doi.org/10.1101/gr.9.2.189).
- [10] Núñez, J., Llacer, J. (2003). Astronomical image segmentation by self-organizing neural networks and wavelets, *Neural Networks*, Vol. 16, No. 3-4, 411-417, doi: [10.1016/s0893-6080\(03\)00011-x](https://doi.org/10.1016/s0893-6080(03)00011-x).
- [11] Chen, E.-L., Chung, P.-C., Chen, C.-L., Tsai, H.-M., Chang, C.-I. (1998). An automatic diagnostic system for CT liver image classification, *IEEE Transactions on Biomedical Engineering*, Vol. 45, No. 6, 783-794, doi: [10.1109/10.678613](https://doi.org/10.1109/10.678613).

- [12] Makadia, A., Pavlovic, V., Kumar, S. (2008). A new baseline for image annotation, In: Forsyth, D., Torr, P., Zisserman, A. (eds.), *Computer Vision – ECCV 2008. ECCV 2008. Lecture notes in computer science*, Vol. 5304, Springer, Berlin, Heidelberg, Germany, doi: [10.1007/978-3-540-88690-7_24](https://doi.org/10.1007/978-3-540-88690-7_24).
- [13] Chen, Z.-Y., Lin, W.-C., Ke, S.-W., Tsai, C.-F. (2015). Evolutionary feature and instance selection for traffic sign recognition, *Computers in Industry*, Vol. 74, 201-211, doi: [10.1016/j.compind.2015.08.007](https://doi.org/10.1016/j.compind.2015.08.007).
- [14] Deng, X., Li, Y., Weng, J., Zhang, J. (2019). Feature selection for text classification: A review, *Multimedia Tools and Applications*, Vol. 78, No. 3, 3797-3816, doi: [10.1007/s11042-018-6083-5](https://doi.org/10.1007/s11042-018-6083-5).
- [15] Baccianella, S., Esuli, A., Sebastiani, F. (2014). Feature selection for ordinal text classification, *Neural Computation*, Vol. 26, No. 3, 557-591, doi: [10.1162/NECO_a_00558](https://doi.org/10.1162/NECO_a_00558).
- [16] Baecchi, C., Uricchio, T., Bertini, M., Del Bimbo, A. (2016). A multimodal feature learning approach for sentiment analysis of social network multimedia, *Multimedia Tools and Applications*, Vol. 75, No. 5, 2507-2525, doi: [10.1007/s11042-015-2646-x](https://doi.org/10.1007/s11042-015-2646-x).
- [17] Chen, J., Wang, T., Gao, X., Wei, L. (2018). Real-time monitoring of high-power disk laser welding based on support vector machine, *Computers in Industry*, Vol. 94, 75-81, doi: [10.1016/j.compind.2017.10.003](https://doi.org/10.1016/j.compind.2017.10.003).
- [18] Guinea, D., Ruiz, A., Barrios, L.J. (1991). Multi-sensor integration—An automatic feature selection and state identification methodology for tool wear estimation, *Computers in Industry*, Vol. 17, No. 2-3, 121-130, doi: [10.1016/0166-3615\(91\)90025-5](https://doi.org/10.1016/0166-3615(91)90025-5).
- [19] Hidalgo-Mompeán, F., Gómez Fernández, J.F., Cerruela-García, G., Márquez, A.C. (2021). Dimensionality analysis in machine learning failure detection models. A case study with LNG compressors, *Computers in Industry*, Vol. 128, Article No. 103434, doi: [10.1016/j.compind.2021.103434](https://doi.org/10.1016/j.compind.2021.103434).
- [20] Edward I, G., Foster, D.P. (2000). Calibration and empirical Bayes variable selection, *Biometrika*, Vol. 87, No. 4, 731-747, doi: [10.1093/biomet/87.4.731](https://doi.org/10.1093/biomet/87.4.731).
- [21] Breiman, L., Friedman, J., Olshen, R.A., Stone, C.J. (1984). *Classification and regression trees*, 1st Edition, Chapman and Hall/CRC, New York, USA, doi: [10.1201/9781315139470](https://doi.org/10.1201/9781315139470).
- [22] Rokach, L. (2010). *Pattern classification using ensemble methods*, World Scientific, Singapore, doi: [10.1142/7238](https://doi.org/10.1142/7238).
- [23] Breiman, L. (2001). Random forest, *Machine learning*, Vol. 45, No. 1, 5-32, doi: [10.1023/A:1010933404324](https://doi.org/10.1023/A:1010933404324).
- [24] Breiman, L. (1998). Arcing classifier (with discussion and a rejoinder by the author), *Annals of Statistics*, Vol. 26, No. 3, 801-849, doi: [10.1214/aos/1024691079](https://doi.org/10.1214/aos/1024691079).
- [25] Hastie, T., Tibshirani, R., Friedman, J. (2009). *The elements of statistical learning, Data mining, inference, and prediction*, Second Edition, Springer, New York, USA, doi: [10.1007/978-0-387-84858-7](https://doi.org/10.1007/978-0-387-84858-7).
- [26] Breiman, L. (1996). Bagging predictors, *Machine Learning*, Vol. 24, No. 2, 123-140, doi: [10.1007/bf00058655](https://doi.org/10.1007/bf00058655).
- [27] Breiman, L. (1999). Pasting small votes for classification in large databases and on-line, *Machine Learning*, Vol. 36, No. 1, 85-103, doi: [10.1023/A:1007563306331](https://doi.org/10.1023/A:1007563306331).
- [28] Okun, O. (2011). *Feature selection and ensemble methods for bioinformatics: Algorithmic classification and implementations*, IGI Global, Hershey, Pennsylvania, USA, doi: [10.5555/2050025](https://doi.org/10.5555/2050025).
- [29] Gelfand, S.B., Ravishanker, C.S., Delp, E.J. (1991). An iterative growing and pruning algorithm for classification tree design, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 2, 163-174, doi: [10.1109/34.67645](https://doi.org/10.1109/34.67645).
- [30] Igual, L., Seguí, S. (2017). *Introduction to data science; A Python approach to concepts, Techniques and applications*, Springer, Cham, Switzerland.
- [31] Carletti, M., Masiero, C., Beghi, A., Susto, G.A. (2019). Explainable machine learning in Industry 4.0: Evaluating feature importance in anomaly detection to enable root cause analysis, In: *Proceedings of 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, Bari, Italy, 21-26, doi: [10.1109/SMC.2019.8913901](https://doi.org/10.1109/SMC.2019.8913901).